Name Youmas Khan
ID 12422
Semester summer II

Q NO 1 => Ans:- A variable provide us with name storage that our program can manipulate. each variable in a java has specific type, which determine the size and layout of the variable memory, the range of values that can be stored within that memory, and the set of operation that can be applied to the variable.

you must declare all variable before they can be used. Following is the basic form of a variable declaration.

data type variable [= value], variable [=value]

Here datatype is the one of java's data types and variable is the name of the variable. to declare more than one variable of the specific type: you can use a comma-seperated list.

example:-

```
int a, b, c                  // Declare three int, a, b, c
int a = 10, b = 10           // example of initialization
byte B = 23;                 // initializes a byte variable B.
double pi = 3.14159;         // declare and assigns a value of pi
char a = 'a';                // the char variable a
                             // is initialized with
                             // value 'a'.
```

* local variables:-

local variable are declared in methods, constructors, or blocks.
local variable are created when the method, constructor or block is entered and the variable will be destroyed once it exits the method. constructor, or block.
Access modifiers cannot be used for local variable.
local varible are visable only within the declared method, constructor, or block.
local variable are implement at stack level internally.
There is no default value for local variable, so local variable should be declared and an initial value should be assigned before the first use.

Example:-

Here age is a local variable. This is define inside pupage() method and its scope is limited to only this method.

```
Public Class Test {
    public void pupAge(){
        int age = 0;
```

```
    age = age + 7;
    System.out.println("Puppy age is: " + age);
  }
  Public static void main (String args[]){
      Test test = new test();
      test.pupage();
    }
  }
```

This will produce the following result,

Puppy age: 7.

* Instance variable:- instance variable are declared in a class but outside a method, constructor or any block.

When a space is allocated for an object in the heap a slot for each instance variable value is created.

instance variable are created when an object is created with the use of the keyword 'new' and destroyed when the object is destroyed.

instance variable hold value that must be referenced by more than one method.

Construtor or block or essential parts of an objects state that must be present throughout the class.

instance variable can be declared in class level before or after use.

Access modifiers can be given for instance variable

The instance variable are visable for all methods. Construtor and block in the class.

Normally, it is recommended to make these variable private (access level). However visibility for subclasses can be given for these variable with the uses of access modifiers.

instance variable have default value. for numbers, the default value is 0, Booleans it is false, and for object reference it is null values can be assigned during the declaration or within the Construtor.

instance variable can be accessed direcly by calling the varible name inside the class.

However, within static method, (when instance variable are given accessibility) they should be called using the fully qualified name.

Example:

```
import java.io.*;
public class employee {
// this instance variable is visable
   for any child class.
public String name;
// salary variable is visiable in
   employee class only;
private double salary;
// the name variable is assigned in
   the constructor.
public Empolyee (String empname) {
    name = empname;
  }
// The salary variable is assigned
   a value.
public void setSalary (double empsal) {
}
// this method prints the employee
   details.
public void printEmp () {
  System.out.println ("name" : " + name);
  System.out.println ("salary: " + salary);
  }
```

Next page

```
}
    public static void main(String args[]){
    Employee empone = new employee("Ramsitha");
    empone.setSalary(1000);
    empone.PrintEmp();
    }
}
```

out put

name : Ramsitha
salary : 1000.0

* Class/static variable:-

class variable also known as static variable are declared with the static keyword in a class, but outside a method constructor or a block.

These would only be one copy of each class variable per class. regardless of how many object are created from it.

static variable are rarely used other then being declared as constant. Constant are variable that are declared public/private, final and static, constant variable never change from their initial value.

static variable can be accessed by calling with the class name, class name, variable name.
When declaring class variable as public static final, then variable names (constant) are all in upper case. if the static variable are not public and final, the naming syntax is the same as instance and local variable.

Example:-

```
import java.io.*;
public class Empolyee {
// salary variable is a private
  static variable
private static double salary;
// DEPARTMENT is a constant
public static final string DEPARTMENT =
  "Development";
public static void main (string args[]){
      salary = 1000;
System.out.println(DEPARTMENT +"average
  salary:" + salary);
  }
}
```

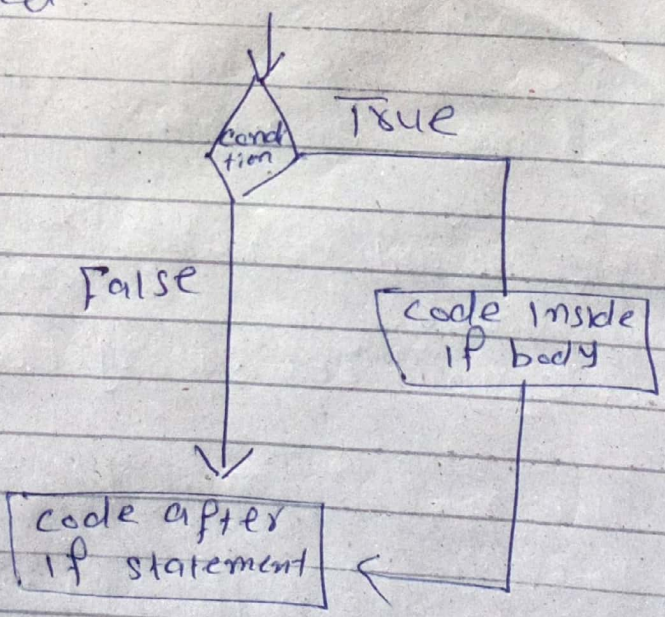
output:- Development average
Salary : 1000

Q.No 2 Ans:- If in java :-

If statement consists a condition, followed by statement or a set of statement as shown below.

```
If (condition) {
    Statement(s);
}
```

The statement gets executed only when the given condition is true. If the condition is false then the statements inside if statement body are completely ignored.



Example of If statement:

```
public class If statement example {
public static void main(String args[]) {
```

(9)

```
        int num=70;
        if (num<100){
    /*this println statement will
only execute
* if the above condition is true
 */
system.out.println("number is less than
                        100");
    }
  }
}
```

output

number is less than 100.

Q.NO3: Answer:-

If-else-If in Java:-

If-else-If statement is used when we need to check multiple condition. in this statement we have only one "if" and one "else", however we can have multiple "else if". It is also known as if else if ladder. This is how it looks.

```
if (condition_1){
/* if condition_1 is true execute
    this*/
statement(s);
```

```
    }
    else if (condition_2){
    /* execute this if condition_1 is
    not met and * condition_2 is met
    */
      statement(s);
    }
    elseif (condition_3) {
    /* execute this if condition_1 & condition_2
    a * not met and condition_3 is met
    */
      statement(s);
    }
    .
    .
    .
      else {
          /* If none of the condition
          is true
          * then these statement gets
          executed
          */
          statements);
      }
```

Note:- The most important point to note here is that in if-else-if statement, as soon as the condition is met, the corresponding set of statement get executed. rest get

ignored. If none of the condition is met then the statement inside "else" gets executed.

Example of if-else-if.

```
Public class IFElseIf example {

  Public static void main (string args[]){
      int num = 1234;
      If(num <100 && num >=1) {
   System.out.println ("its a two digit number");
   }
    else if (num <1000 && num>=100) {
    system.out.println ("its a three digit no");
   }
   else if (num <10000&& num>=1000) {
   System.out.println ("its a four digit no");
   }
    else if (num< 100000 && num>=10000) {
   system.out.println(" its a five digit no");
   }
   else{
    System.out.println(" number is not between
                                    1 & 9999");
      }
   }
}
```

output:

its a four digit number.

Q NO4: Answer: loops are used to execute a set of statements repeatedly until a particular condition is satisfied. in java we have three types of basic loops. for, while and do while. in this tutorial we will learn how to use "for loop in java.

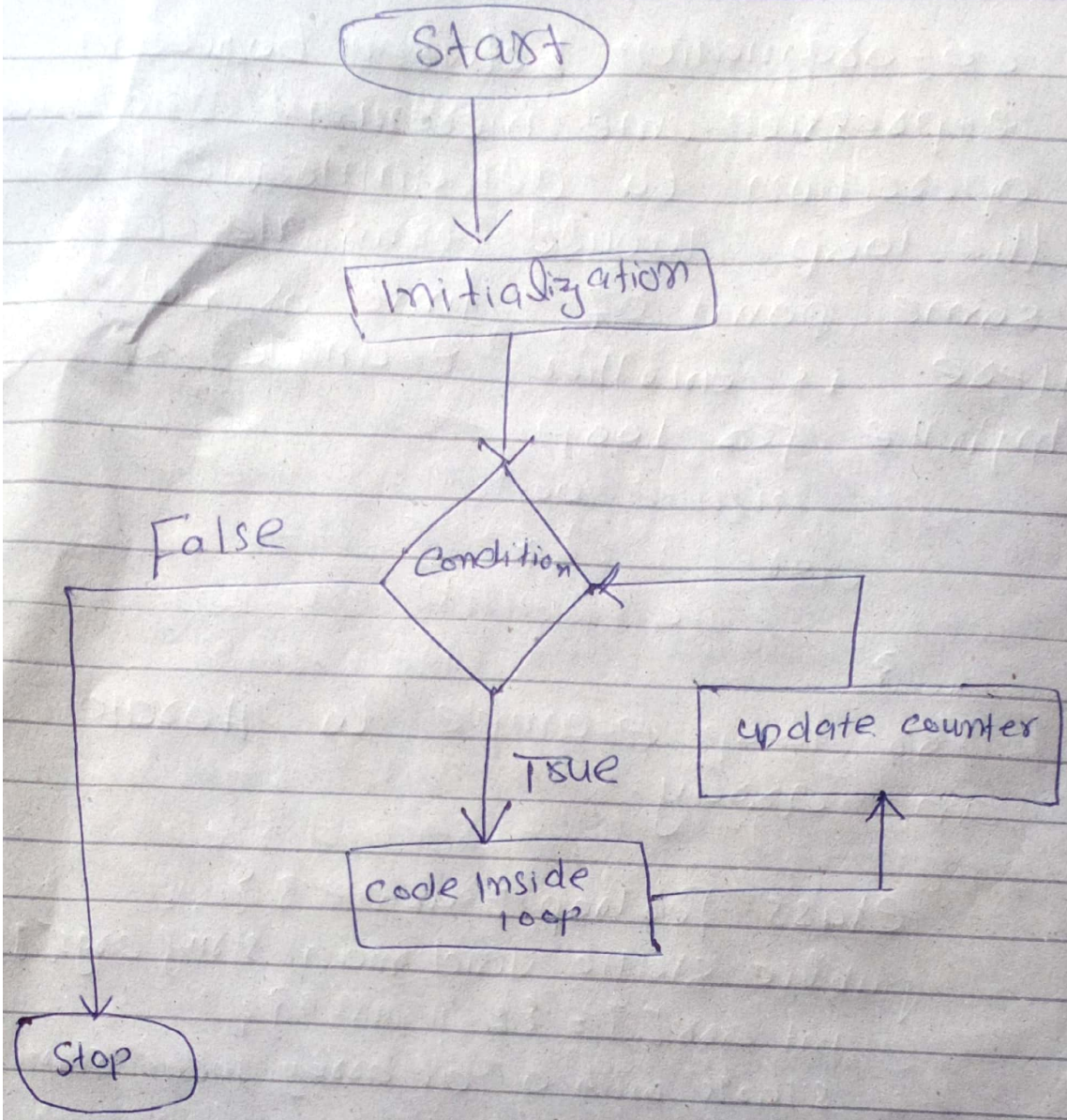
Syntax of for loop:

```
for (initialization; condition;
      increment/decrement)
    {
        statement(s);
    }
```

flow of execution of the for loop.

As a program executes, the interpreter always keeps track of which statement is about to be executed. we call this the control flow, or the flow execution of the program.

Next page.



First step: In for loop, initialization happens first and only one time. Which means that the initialization part of for loop only executes once.

Second step: condition in for loop is evaluated on each iteration. If the condition is true then the statements inside for loop body gets executed.

once the condition returns false, the statements in for loop does not execute and the control gets transferred to the next statement in the program after the loop.

Third step :- After every execution of for loop's body the increment / decrement part of for loop executes that updates the loop control.

Fourth step :- After third step, the control jumps to second step and condition is re-evaluated.

Example of simple for loop :-

```
class Forloop Example {
    public static void main (String args[]
                                    ){
        for (int i=10; i--) {
    System.out.println(" The value of i is
                                    "+i);
        }
    }
}
```

output :-

The value of is : 10

" " " : 9

" " " : 8

In the above program:
int i=1 initialization expression
i>1 is condition (Boolean expression)
i- decrement operation

Infinite for loop:
The important of Boolean expression and increment/decrement operation co-ordination.

Example: class forloopexample2 {

Public static void main (string args[]){
for (int i=1; i>=1; i++){
System.out.println("The value of i is:"+i);

}
}
}

This is an infinite loop as the condition would never return false. The initialization step is setting up the value of variable i to 1, since we are incrementing the value of i, it would always be greather than 1 the boolean expression: i>1) so it would never return false. This would eventually lead to the infinite loop condition. Thus it is important to see

co-ordination between boolean expression and increment/decrement operation to determine weather the loop would terminate at some point of time or not.
Here is another example of infinte for loop.

```
// infinte loop
for (; ;) {
    // statement(s)
}
```

For loop example to iterate an array

```
class ForLoopExample {
public static void main(String args[]){
    int arr[] = {2,11,45,9};
// i starts with 0 as array index starts
    with 0 too
for (int i =0; i<arr.length; i++){
    System.out.println(arr[i]);
    }
  }
}
```

output

2
11
45
9

* Enhanced for loop :-

enhanced for loop is useful when you want to iterate Array/ Collection. It is easy to write and understand.
lets take the same example that we have written above and rewrite it using enhanced for loop.

```
class for loop example 3 {
public static void main(string args[]){
    int arr[] = {2,11,45,9};
    for (int num : arr){
     system.out.println(num);
    }
  }
}
```

Note: In the above example, I have declared the num as int in the ~~element~~ enhanced for loop. This will change depending on the data type of array. For example the enhance for loop for string type would look like this.

```
string arr[] = {"hi", "hello", "bye"};
for (string str : arr){
    system.out.println(str);
}
```

While loop in java with example:-

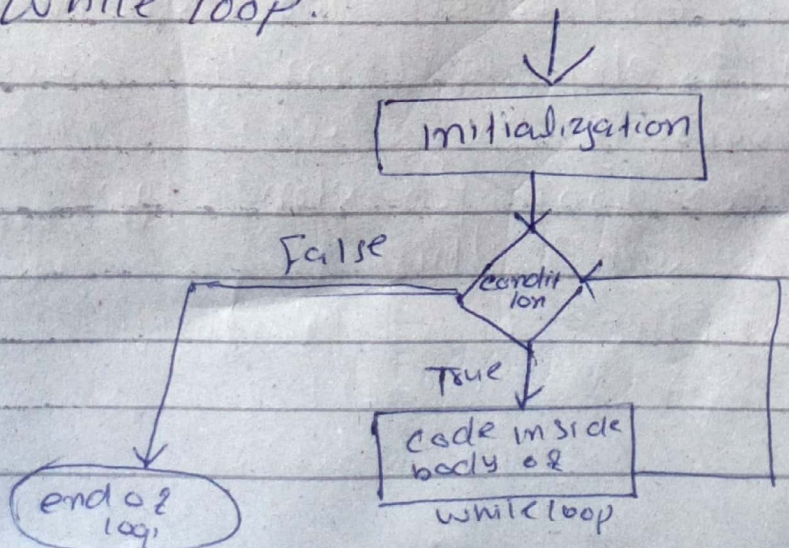
In the last tutorial we discussed for loop. In this tutorial we will discuss while loop. While loop is used to execute a set of statement repeatedly until a particular condition is satisfied.

Syntax of while loop:

```
while (condition)
{
    statement(s);
}
```

How while loop works

In while loop, condition is evaluated first and if it return true then the statements inside while loop execute. When condition return false the control comes out of loop and jumps to the next statement after while loop.



While loop example:

```
class whileloop example {
public static void main (string args[]) {
    int arr [] = {2, 11, 45, 9};
// ; starts with 0 as array index
        starts with 0 too
    int i = 0
    while (i<4) {
        System.out.println(arr[i]);
            i++;
    }

  }

}
```

output

2
11
45
9

* do-while loop in Java with example:

do while loop is similar to while loop, however there is a difference between them. in while loop, however condition is evaluated before the execution of loop's body but in do-while loop condition is evaluted

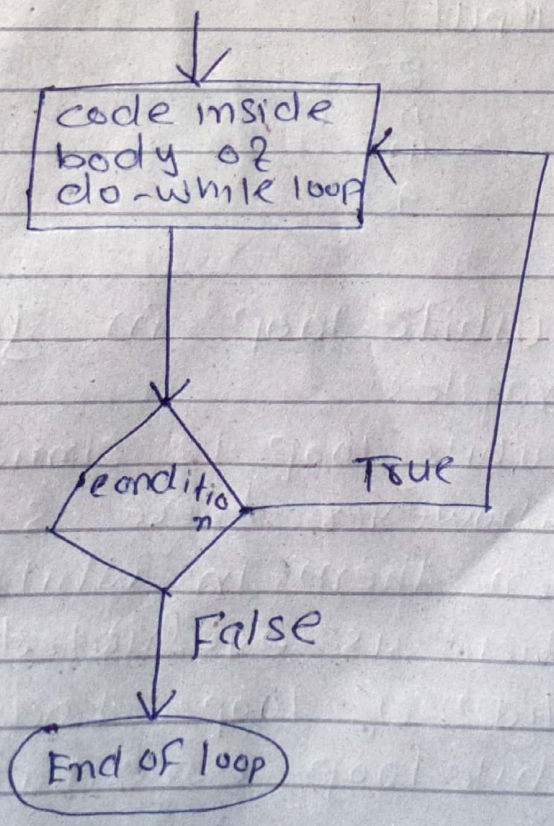
after the execution of loop's body.

Syntax of do-while loop:

```
do
{
    statement(s);
} while (condition);
```

How do-while loop works

First the statement inside loop execute and then the condition gets evaluated if the condition returns true then the controls get transferred to the "do" else it jumps to next statement after do while.



do-while loop Example:

```
class DowhileloopExample {
    public static void main(String args[]){
        int i =10;
        do{
            System.out.println(i);
            i--;
        } while (i>1);
    }
}
```

Output:

```
10
9
8
7
6
5
3
4
2
```

Q1\05 : Answer:- increment(++) and decrement(--) operators in java programing let you easily add 1 to or subtract 1 from a variable. for example, using increment operator you can add 1 to a variable,

a++;

An expression that uses an increment or decrement operator is a statement

itself. that's because the increment or decrement operator is also a type of assignment operator. because it changes the variable it applies to.

you can also use an increment or decrement operator in a assignment statement.

```
int a = 5
int b = a--;  // both a and b are set to 4
```

Increment and decrement operator can be placed before (prefix) or after (postfix) the variable they apply to. If you place an increment or decrement operator before its visable. the operator is applied before the rest of the expression is evaluated. If you place the operator after the variable the operator is applied after the expression is evaluated.

for example:

```
int a = 5;
int b = 3;
int c = a * b++; // c is set to 15
int d = a* ++b; // d is set to 20
```

```
class incDec{
  public static void main(String args[]){
    int a = 1;
    int b = 2;
    int c;
    c = ++b;
    b = a++;
    c++;
    System.out.println("a =" +a);
    System.out.println("b =" + b);
    System.out.println("c =" +c);
  }
}
```

output

a = 2
b = 3
c = 4